Bayesian networks for incomplete data analysis in form processing

Emilie Philippot, Santosh K.C., Abdel Belaïd, Yolande Belaïd

Bayesian Networks for Incomplete Data Analysis in Form Processing*

Emilie Philippot[†,♮], K.C. Santosh[†,‡,♮], Abdel Belaïd[†], Yolande Belaïd[†]

[†]Université de Lorraine – LORIA
BP 239 – 54506 Vandoeuvre-lés-Nancy Cedex, France

[‡]Communications Engineering Branch, US National library of Medicine (NLM)
National Institutes of Health (NIH), 8600 Rockville Pike, Bethesda, MD 20894, USA

Email: emilie.philippot@gmail.com, santosh.kc@nih.gov and {abdel.belaid, yolande.belaid}@loria.fr

[♮]**Corresponding authors.**

Abstract

In this paper, we study Bayesian network (BN) for form identification based on partially filled fields. It uses electronic ink-tracing files without having any information about form structure. Given a form format, the ink-tracing files are used to build the BN by providing the possible relationships between corresponding fields using conditional probabilities, that goes from individual fields up to the complete model construction. To simplify the BN, we sub-divide a single form into three different areas: header, body and footer, and integrate them together, where we study three fundamental BN learning algorithms: Naive, Peter & Clark (PC) and maximum weighted spanning tree (MWST). Under this framework, we validate it with a real-world industrial problem i.e., electronic note-taking in form processing. The approach provides satisfactory results, attesting the interest of BN for exploiting the incomplete form analysis problems, in particular.

Index Terms

Bayesian Networks, Electronic Note-taking, Form Processing.

1 Introduction

1.1 Context

This work is done in direct collaboration with the *Actimage* company who has been investigating technological innovations through on-line interfaces like pens and tablets. The global aim is to complete the partially filled form in accordance with the clients. This is however, common in some applications like sketching interfaces in which the user sketches a few partial schemas and the system then adjusts the ratings, scales and completes the missing parts. In this paper, we focus on business form processing. Purchase orders and control forms like an inspection health service, an inventory and a site record, are two different types of examples. In order to provide an intuitive feeling, in the very beginning, we start with an example showing the partially filled form in Fig. 1. In this illustration, on the left, an overall form (structure plus partially filled

*This work has been conducted under a CIFRE agreement. Thanks to the Actimage company for collaboration and for learning database that are used in our study.

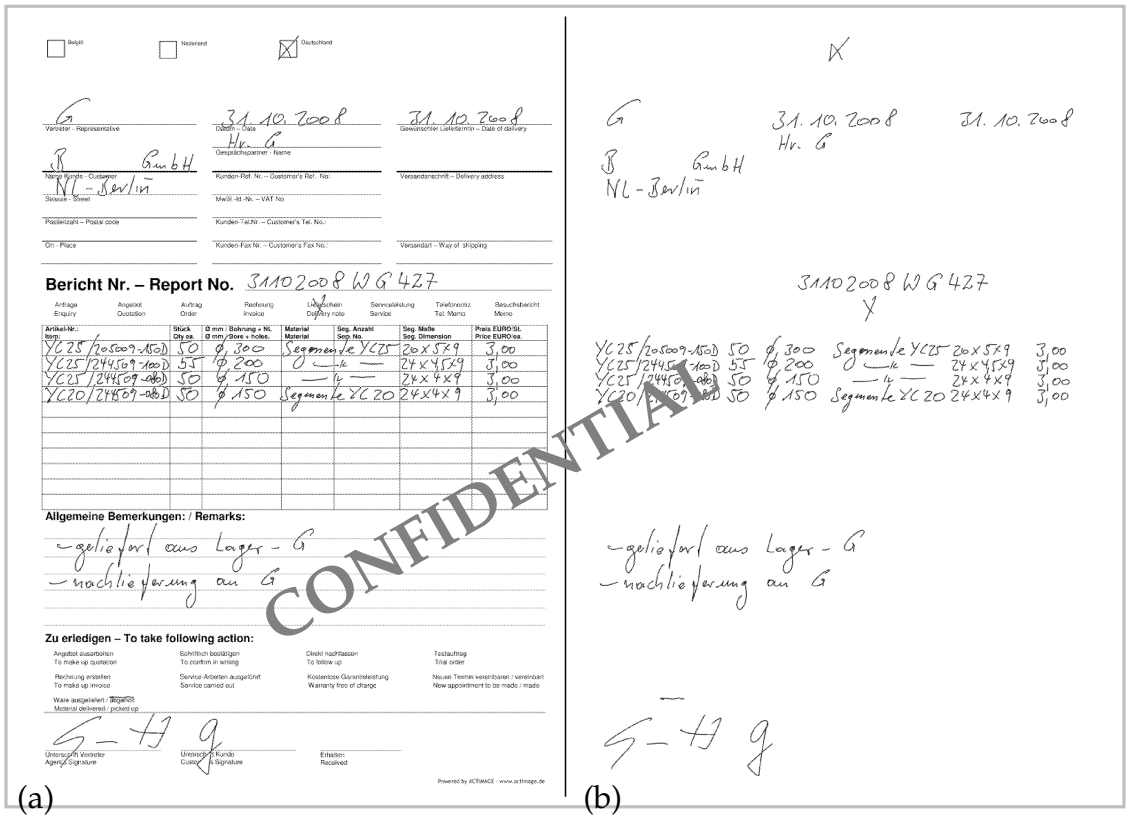


Fig. 1. An example showing (a) the partially filled form where (b) ink-tracing files i.e., input data are separated.

fields) is shown and the input data via electronic ink is shown on the right. As said before, in form processing, fields are partially filled i.e., a few fields are filled but, in some cases, one can get all information about a client or only identification number. Having complete information is however, not happened in reality.

To handle form processing, professionals are basically used to fill the fields by using electronic pen that yields ink-tracing files. These files are then used to parse in order to determine the corresponding form format and of course, the associated class. Globally, our approach can be explained as follows.

1) We first decompose a single form mainly into three different areas of interest. Conventionally speaking, areas of interest are header, body and footer.
2) In each area of interest, corresponding fields (using electronic ink-tracing files) are used to build Bayesian sub-network (BsN). These BsNs from all areas are then integrated together to represent the whole form.
   Such a decomposition simplifies the complexity in matching problem.

In this paper, we have studied three major Bayesian learning algorithms such as Naive, PC and MWST. To validate our approach, an interesting real-world industrial application i.e., electronic note-taking is taken.

1.2 Organisation of the paper

We organise the rest of the paper as follows. An overview of pertinent literature is given in Section 2, followed by a brief explanation of the BN in Section 3. We explain the proposed approach in Section 4, which mainly includes ink-tracing files via electronic pen, form description and its format, areas of interest and BN representation. The approach has been implemented and validated with a real-world application in Section 5 i.e., electronic note-taking in form processing. In our implementation, we start with explaining field extraction, BN learning and recognition process. Full experimental results are reported in Section 6. It includes dataset and evaluation protocol, and test results. The test results are followed by the discussion in Section 7. The paper is concluded in Section 8.

2 Related work

Research on form classification has an extremely rich state-of-the-art literature but, the use of BN has not been noticeably appeared. However, using BN is not a novel concept. Under this purview, we are limited to handwriting and document analysis and or classification using BNs, aiming to attest the interest of it in the domain. After that, we will highlight very recent works on Bayesian classifiers.

In character recognition, several approaches provide BN's ability to accurately identify the structure of the character by using dependency relationships between the segmented components i.e., strokes, for instance. In other words, relative positioning between the strokes provides a key element to exploit the structure of the complete character. As an example, in [Cho and Kim, 2003], for Korean Hangul characters, a hierarchy of components is proposed for character modelling where a syllable model, grapheme models, stroke models and point models are comprehensively studied. Each model is constructed with sub-components and their relations i.e., local dependencies. In [Verron *et al.*, 2007], authors proposed similar system for on-line isolated character recognition, where any character is modelled based on stroke models and their spatial relations. A dynamic BN (DBN) (DBNs are the extension of 1D hidden Markov models (HMMs) which can handle several observations and state sequences) is used to exploit dependencies between strokes. One of the major advantages of the system lies in that they are against geometric variations and are having sufficient information to distinguish characters. It is mainly because of the fact that the set of spatial relations do not change as long as we have set of strokes since relations are relative in nature. Concerning importance and effectiveness of spatial relations between the strokes, we refer to the very recent study presented in [Santosh *et al.*, 2012]. A more refined model can be achieved by exploiting the correlations between variables. In this context, we find the work of Hallouli et al. [Hallouli *et al.*, 2002] where probabilistic models based on dynamic BNs are developed for off-line handwritten character recognition. It uses 2D models by integrating two HMMs to develop a BN. The first HMM model is obtained from pixel observation in columns (i.e., vertical-HMM), the second observation from lines (i.e., horizontal-HMM). This model overcomes the limitations of HMMs with an optimised model by integrating pixel's information and line-based observations. However, the choice of coupling links must be studied via structure learning, for example. Likforman-Sulem and Sigelle in [Likforman-Sulem

and Sigelle, 2008, 2009] proposed another approach called auto-regressive HMMs (AR-HMMs) that provides better performance.

Another domain where BNs have been investigated is natural language processing (NLP). As an example, in [Piwowarski *et al.*, 0002], the authors combine text and document structure for document categorisation. Based on this, a document is sub-divided into three main levels: the document in total, pages and sections. As a consequence, BN can now represent the hierarchy in a single frame. In [Weissenbacher, 2006; Weissenbacher and Nazarenko, 2011], the focusing point is to identify anaphoric pronouns in heterogeneous data. According to them, in the domain of automatic language processing, knowledge can be based on either a linguistic or surface showings. BNs allow them to merge these two data types, in terms of surface indices and linguistic elements. In this study, BN holds better performance in comparison to the basic conventional approaches.

Another research domain where BNs have been increasingly used is document structure indexing i.e., document navigation on the Web and in large databases of historical documents. A document image is represented using areas of interest such as titles, sections and paragraphs. These components are linked together in a structure where the relationships can be described by conditional dependencies and therefore formalised via BNs. The major approaches describing the spatial relations, are symbolic projection (2D String), graph-based representation such as trees and attributed relational graphs (ARG). BNs as a directed acyclic graphs (DAG) is used to represent relations and conditional probability to express uncertainty on the relationships. In this framework, the task is to realise how BNs can be adapted in accordance with the problem of classification or indexing.

In document classification, the complexity of the physical layouts makes the analysis complex, both for text block extraction and logical component identification. The feature labelling that depends on inherent instability of the physical structures can directly affect the logical level. A typical example of such documents are table of contents in periodicals or magazines [Belaïd, 2001]. In [Souafi-Bensafi *et al.*, 2002], a generic probabilistic model is used for logical labelling of text blocks in documents, using BN classifiers where a prototype has been implemented and applied to periodical magazines. In their comparative study, it is found that the learnt BNs do not make any significant difference over naive BN. It is because of the nature of their data which is particularly well represented by naive structures. For documents containing both text and graphics, we refer to work [Mahjoub and Jayech, 2010] where it uses variants of BN such as: naive (NN), naive augmented by a tree (TAN) [Friedman *et al.*, 1997] and naive augmented by a forest (FAN) [Jiang *et al.*, 2005]. A TAN is a tree where all leaves are connected. This is due to the conditional independence assumption in naive Bayes. It however, produces poor probability estimation [Friedman and Goldszmidt, 1996]. One way to alleviate this assumption is to extend the structure to explicitly represent variable dependencies by adding arcs between them. These additional arcs are now useful to maximise the weights via MWST algorithm. The FAN tree is obtained from the TAN tree by removing arcs whose mutual information is below a certain threshold. This is done to remove the conditional dependencies which are not enough representative.

Document indexing where BNs are employed, is certainly an interesting domain. It is mainly focused on searching and analysing semi-structured and XML-like documents. Here, the task is related to handle both content and structure that helps to localise the specific information embedded in the document. Unlike the past studies operated in plain document without considering its structure, document content structure has been changed including the vision. In addition to XML representation, document structure is enriched by meta-data that provides the description of heterogeneous information. In [Sebastiani, 2002], an example of is illustrated by using different aspects, but research is still on-going. L. Denoyer and P. Gallinari [Denoyer and Gallinari, 2004] proposed a generative MN model for document structure. The model is able to take structure into account and the information content about its type. To optimise the computational complexity and to allow robust parameter estimation, they are restricted to simple models via local structural dependencies exploitation.

Besides, very recently, authors highlight feature selection issues and performance of the Bayesian classifiers. In [Subrahmanya and Shin, 2013], authors focus on grouping of features during model development and the selection of a small number of relevant groups. They aim to improve the interpret-ability of the learnt parameters and to avoid parameters which are basically manually tuned. Multi-instance (MI) learning is another interesting work, where learning examples are represented by a bag-of-instances instead of a single instance [Jiang *et al.*, 2013]. In this work, authors propose Bayesian-KNN (BKNN) and Citation-KNN (CKNN) to solve multi-instance classification problems, where voting is based on the weighted distance. In [Wang *et al.*, 2014], a non-naive Bayesian classifier (NNBC) is proposed in which the independence assumption is removed and the marginal probability density function estimation is replaced by the joint probability density function estimation, in order to achieve satisfactory performance of the classifiers. It includes a new technique to estimate the class-conditional probability density function based on the optimal bandwidth selection. In that framework, an interesting application i.e., simultaneous fault diagnosis is proposed [He *et al.*, 2014], where authors propose a new model of Bayesian classifier that is able to remove the independence among the features. As said before, it basically uses an optimal bandwidth selection to estimate the class-conditional probability density function.

3 Materials

3.1 Basics on BN

We repeat that the BNs have been increasingly used in the community of document analysis and data mining [Kebairi *et al.*, 1998]. BNs, also known as belief networks (or Bayes nets for short), belong to the family of probabilistic graphical models (PGMs). These graphical structures are used to depict conditional independence among random variables in the domain and encodes the joint probability distribution [Pearl, 1988; Wong and Leung, 2004]. In other words, BN is the intersection between graph theory and probability [Jensen, 1996; Naïm *et al.*, 2007]. Basically, there are three types of graphical probabilistic models based on their structure:

1) the directed acyclic graph (DAG) with oriented arcs;
2) the Markov random field (MRF) with undirected arcs; and

3) the chains of graphs that are composed at the same time of directed and undirected arcs.

A BN is defined by a directed acyclic graph $G = (V, E)$, where V is the set of nodes and E the set of arcs. In particular, each node in the graph represents a random variable, while the edges between the nodes represent probabilistic dependencies among the corresponding random variables.

Let X is a Bayesian network with respect to G if its joint probability density function (with respect to a product measure) can be written as a product of the individual density functions, conditional on their parent variables [Russell and Norvig, 2003]:

$$p(x) = \prod_{v \in V} p\left(x_v | x_{\mathrm{pa}(v)}\right), \tag{1}$$

where $\mathrm{pa}(v)$ is the set of parents of v (i.e,. those vertices pointing directly to v via a single edge), and $X = (X_v)_{v \in V}$ be a set of random variables indexed by V. Therefore, for any set of random variables, the probability of any member of a joint distribution can be calculated from conditional probabilities using the chain rule (given a topological ordering of X) as follows:

$$P(X_1 = x_1, \ldots, X_n = x_n) = \prod_{v=1}^{n} P\left(X_v = x_v | X_{v+1} = x_{v+1}, \ldots, X_n = x_n\right), \tag{2}$$

It can then be compared with Eq. (1),

$$P(X_1 = x_1, \ldots, X_n = x_n) = \prod_{v=1}^{n} P\left(X_v = x_v | X_j = x_j \text{ for each } x_j \text{ which is a parent of } X_v\right), \tag{3}$$

To develop a BN, we basically first create a graph G. We then ascertain the conditional probability distributions of each variable given its parents in G. In many cases, in particular in the case where the variables are discrete, if we define the joint distribution of X to be the product of these conditional distributions, then X is a Bayesian network with respect to G [Neapolitan, 2004]. Consider x_1, x_2 and x_3 be three variables, representing for example the fields: "Country", "Zip code" and "Town" respectively in the address area of the provided form. There are three types of possible relations between them viz. serial, convergent and divergent.

1) In a serial connection, x_1 is connected to x_2 via x_3 i.e., $x_1 \rightarrow x_2 \rightarrow x_3$.
2) In a divergent connection, x_1 and x_2 are dependent i.e., $x_1 \leftarrow x_3 \rightarrow x_3$.
3) In a convergent connection, x_1 and x_2 are independent, and x_3 is dependent on them i.e., $x_1 \rightarrow x_3 \leftarrow x_3$.

Therefore, two nodes x_1 and x_2 are d-separated if, for any path (undirected) between them, there is an intermediate variable x_3 which is in the form of

- either serial or divergent connection, and x_3 is known
- or convergent connection, and neither x_3 nor any of x_3's descendants are known.

If nodes x_1 and x_2 are d-separated by x_3, then x_1 and x_2 are conditionally independent given x_3. Likewise, nodes x_1 and x_2 are said to be d-connected if they are not d-separated. For example, X is a BN with respect to G if, for any two nodes x_1 and x_2: $X_{x_1} \perp\!\!\!\perp X_{x_2} \,|\, X_z$, where z is a set which d-separates x_1 and x_2.

3.2 BN learning and classification

In this section, we provide an idea about how we perform learning and classification. Learning basically provides how we find the graph structure that best represents the problem based on the parameters computation via estimated conditional probabilities. In the literature, three commonly used algorithms are Naive, PC and MWST. In what follows, we first discuss Naive Bayes algorithm and classification in detail, which is then followed by a quick functioning principle of remaining algorithms.

1) **Naive.** In short, the probability model for a classifier is a conditional model, $p\left(C|F_1,\ldots,F_n\right)$ over a dependent class variable C with a small number of outcomes or classes, conditional on several feature variables F_1 through F_n. In case, the number of features is large, we then reformulate the model to make it more tractable. Using Bayes' theorem, this can be written

$$p\left(C|F_1,\ldots,F_n\right) = \frac{p(C)p(F_1,\ldots,F_n|C)}{p(F_1,\ldots,F_n)}. \tag{4}$$

In practice, since the denominator does not depend on feature values (thus effectively constant) the numerator part is of interest. The numerator is equivalent to the joint probability model, $p(C,F_1,\ldots,F_n)$. This can be expressed by using the chain rule,

$$\begin{aligned} p\left(C,F_1,\ldots,F_n\right) &= p(C)p(F_1,\ldots,F_n|C), \\ &= p(C)p(F_1|C)p(F_2,\ldots,F_n|C,F_1), \\ &= p(C)p(F_1|C)p(F_2|C,F_1)p(F3,\ldots,F_n|C,F_1,F_2), \\ &= p(C)p(F_1|C)p(F_2|C,F_1)\ldots p(F_n|C,F_1,F_2,F_3,\ldots,F_{n-1}). \end{aligned} \tag{5}$$

Now the "naive" conditional independence assumptions come into play: assume that each feature F_i is conditionally independent of every other feature F_j for $j \neq i$ given the category C. This means that $p(F_i|C,F_j) = p(F_i|C), p(F_i|C,F_j,F_k) = p(F_i|C)$, and so on, for $i \neq j,k,l$ and so the joint model can be expressed as

$$\begin{aligned} p(C|F_1,\ldots,F_n) &\propto p(C,F_1,\ldots,F_n) \\ &\propto p(C)p(F_1|C)p(F_2|C)p(F_3|C)\ldots \\ &\propto p(C)\prod_{i=1}^{n} p(F_i|C). \end{aligned}$$

This means that under the above independence assumptions, the conditional distribution over the class variable is

$$p\left(C|F_1,\ldots,F_n\right) = \frac{1}{K}p(C)\prod_{i=1}^{n} p\left(F_i|C\right), \tag{6}$$

where K (the evidence) is a scaling factor dependent only on $F_1,\ldots,F_n$ i.e., a constant if the values of the feature variables are known.

For any training data containing a continuous attribute x, we basically segment the data

by the class and compute the mean and variance of x in each class. Then, the probability density of some value given a class,

$$P\left(x=v|c\right)=\frac{1}{2\pi\sigma_c^2}exp\left(-\frac{(v-\mu_c)^2}{2\sigma_c^2}\right), \tag{7}$$

where μ_c and σ_c^2 respectively be the mean and the variance of the values in x associated with class c.

Based on the independent feature model i.e., the naive Bayes probability, the naive Bayes classifier combines this model with a decision rule. One basic rule is to take the hypothesis which is the most probable based on the maximum a posteriori or MAP decision rule. A Bayes classifier is the function defined as follows,

$$\text{classify}(f_1,\ldots,f_n)=\arg\max_c p(C=c)\prod_{i=1}^{n}p\left(F_i=f_i|C=c\right). \tag{8}$$

Basically, the variables are conditionally independent to the class. In this network, the classification is done using the Bayesian rule. Many studies [Langley *et al.*, 1992; François and Leray, 2006]) have demonstrated the effectiveness of this classifier compared to other structures of BNs. However, this efficiency is highly related to the conditional independence between variables and the simplicity of the problem.

2) **PC.** Peter & Clark (PC) is a search algorithm of conditional independence [Spirtes *et al.*, 2001]. We basically start with a completely connected graph. For each pair of random variables connected by an arc, the algorithm examines the existing conditional independence using the χ^2. Based on it, we remove the corresponding arc. It then checks the conditional independence for the remaining variables in a set until all the conditional independences are removed. Once all the conditional independences detected, it looks for V-Structures to direct arcs.
3) **MWST.** The algorithm maximum weight spanning tree (MWST) is a part of the family of algorithms based on a score [Chow and Liu, 1968]. The goal is to find the tree that goes through all nodes: $x_1,\ldots,x_n$ in the network by maximizing a score defined for all possible arcs. The starting point of the algorithm is a set of n trees, each node representing each variable. Then the trees are merged according to the arc weights.

Generally speaking, in order to fully specify the BN and thus fully represent the joint probability distribution, it is necessary to specify for each node x_i the probability distribution for x_1 conditionally upon $\text{pa}(x_1)$ parents. Often, these conditional distributions include parameters which are unknown and must be estimated from data, sometimes using the maximum likelihood approach,

$$p\left(x_i=st_k|\text{pa}(x_i)=st_j\right)=\frac{N_{i,j,k}}{\sum_k N_{i,j,k}}, \tag{9}$$

where $N_{i,j,k}$ is the number of events in the database where the random variable x_i is in the state st_k and its parents are in the configuration st_j. In case we have very small (or limited) training data, direct probability computation may provide probabilities of 0 or 1. These extreme

probabilities are "very strong" and may affect the decision process. In such a case, one can use Laplace estimator [Jiang *et al.*, 2007]. In this paper, we use maximum likelihood estimator.

To use the BN on new data for its classification, the inference mechanism is employed. It computes the probability of a non-instantiated variable (such as a new completed form) depending on other variables instantiated for a BN. One of the main inference algorithms is that the junction tree, has been developed by [Jensen, 1996; Jensen *et al.*, 1990]. It uses the obtained graph in the learning phase and breaks down into several steps:

1) moralisation,
2) triangulation and
3) search for a maximum spanning tree, which we refer to as junction tree.

Until now, we have discussed the principle of BN. In what follows, we will discuss how we are employing it for form analysis problem.

4 Form processing using BN

4.1 Outline

Developing a template-free form recognition while considering noisy images by recognising target characters is not an easy task in addition to the ambiguous alignment layout [Hirayama *et al.*, 2011a]. This concludes that it does not offer deployment, commercially speaking. Instead, use of handwritten digital ink for form processing is interesting [Tran *et al.*, 2010]. Under this purview, globally speaking, our proposed concept uses ink-tracing files via electronic pen from the professionals. These ink-tracing files do not itself provide any knowledge about the form structure. Therefore, they are now integrated with the given form format as an input to the system. Forms are then decomposed into three major areas of interest i.e., header, body and footer. As soon as we have specified areas of interest, the corresponding BN is built which are then integrated into a single global network. In order to represent the whole form, arcs are inserted between the areas of interest. In Fig. 2, an overall idea of the learning concept is depicted.

Each area has its own significance and allows dynamic reduction in computational complexity of the system since number of fields in each area can be varied. In other words, it simplifies the networks but requires an intelligent data discretisation to optimise the performance. For recognition, the extracted fields are basically served to feed the BN and to allow the inference.

In the following, we will explain the major elements that are used to represent form via BN. We start with highlighting why electronic pen is employed. We then provide details about how global BN for a single form is built. It mainly concerns three areas of interest, including the form structure.

4.2 Electronic pen

Without a surprise, several devices exist for note-taking on screens or tablets. Therefore, instead of relying on this type of touch screen technology, investigating the use of electronic ink on the paper will be accepted for commercial purpose since it offers natural usability.

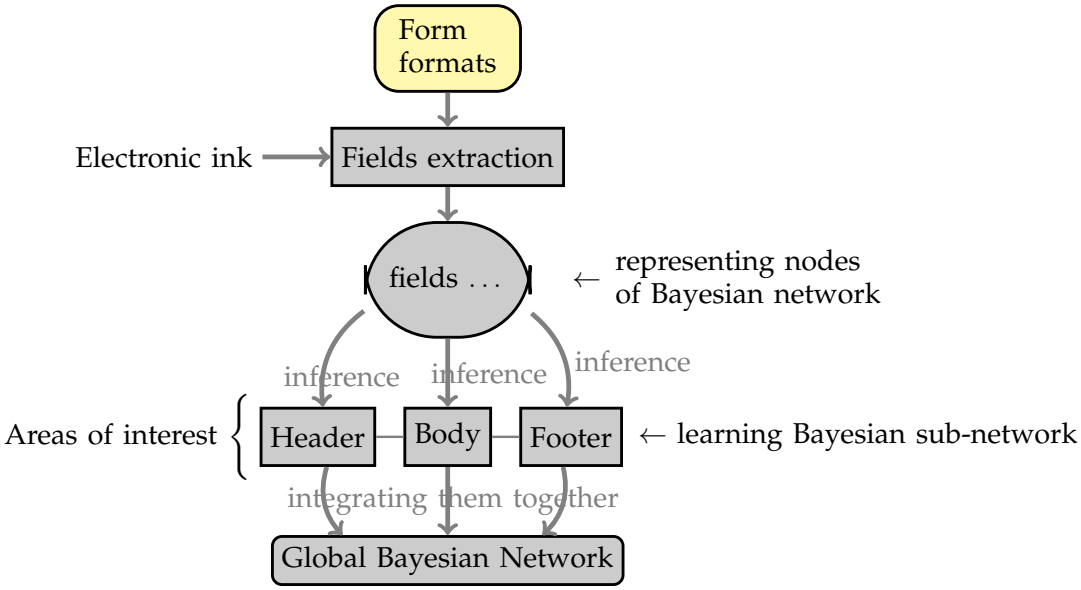


Fig. 2. A quick screen-shot of the learning concept in our proposed method.

Among several types of electronic pens, currently, we use the ANOTO concept. In this concept, the pen equipped with a camera, films the route of the pen where it is possible to accurately detect the writing coordinates and then rebroadcasts the tracings. This technology offers a high reliability as the frame of the paper can provide all the information needed for further processing.

4.3 Form description and its format

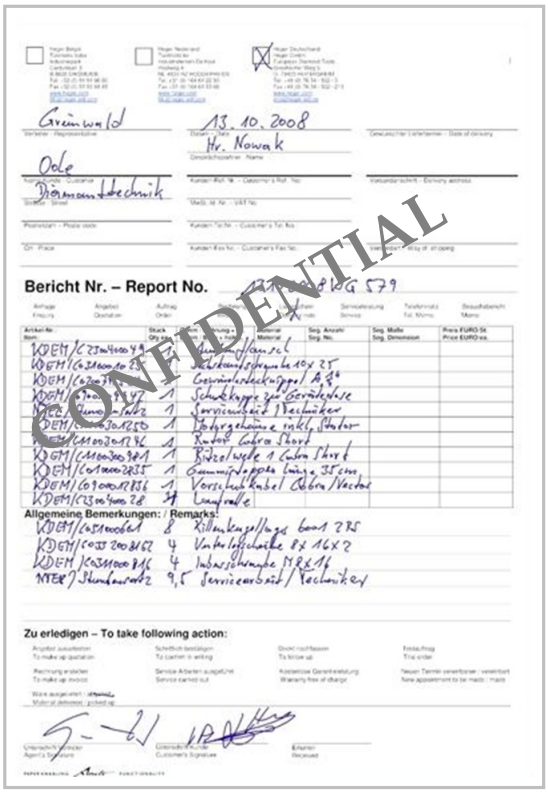
As said before, forms are provided by the *Actimage* company. In Fig. 1 in pp. 2, a sample is provided. Forms contain check boxes, text boxes and free text boxes. All forms are in A4 ($21cm \times 29.7$ cm), in portrait orientation. In average, the number of fields are ranging from 100 and 360 fields in a single form. A field is an elementary area for data input (i.e., ink-tracing files). It will represent a variable v_i as discussed in Section 3.

For each form class, XML representation of its format is provided. Each area of the form is described with its fields. As an example, XML representation of the field ☐ Account is,

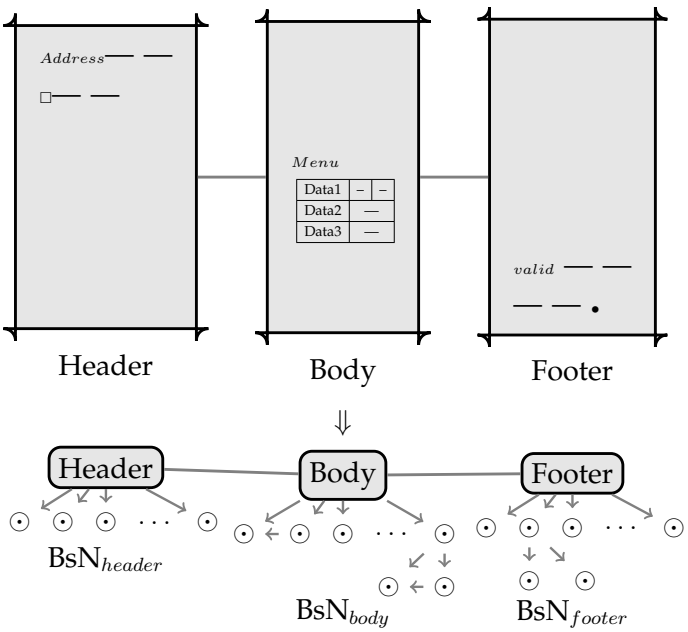
```
<Fields>
<X>21.5 24.5</X>
<Y>148.5 151.5</Y>
<Type>Case_A_Checkmark</Type>
<Label>Account</Label>
</Fields>.
```

For each field, we consider the following elements:

1) *bounding box* – a margin of a few pixels is included in order to absorb overlapping fields due to font variations in handwriting;
2) *type* – it can be either check box, text box or drawing figures;
3) *label* – the predefined labels such as “Mr.” “Mrs.”, “Address”, “Order Number”, for example; and



(a) A sample image showing three areas of interest.



(b) Bayesian sub-network (BsN) for different areas of interest. Each ⊙ represents field of the corresponding area.

Fig. 3. Overall Bayesian network structure for a single form having three different areas with their Bayesian sub-network. To be more illustrative, $\mathrm{BN}_{form} = \{\mathrm{BsN}_{header}, \mathrm{BsN}_{body}, \mathrm{BsN}_{footer}\}$.

4) *areas of interest* – it can be either header, body or footer.

The form format is used to validate the class by checking the consistency between the field types and the data entered. For example, we can check whether an area is actually composed of digits.

4.4 Areas of interest and Bayesian network representation

As said before, to simplify the complexity of the problem, the form is divided into three areas of interest: header, body, and footer. As an example, customer identity, order and its validation represent header, body and footer respectively. Following Fig. 3, since we have three sub-divisions, there exists three Bayesian sub-networks (BsNs) representing header, body and footer separately. In this illustration, we have shown a single arc that connects from one BsN to another in order just to provide an intuitive idea on it. But practically, nodes from every BsNs are connected to each other. In other words, intra-node connections are computed first that exist within the BsNs and then inter-node connections in between the BsNs.

During training, all BsNs of the corresponding forms are integrated into a single BN named a global BN (GBN) where the arcs give the conditional dependence between the BsNs. It is important to notice that the BsN integration is made by taking several forms, not just limited to a single form. As a reminder, the random variables of BsN represent the form fields (denoting

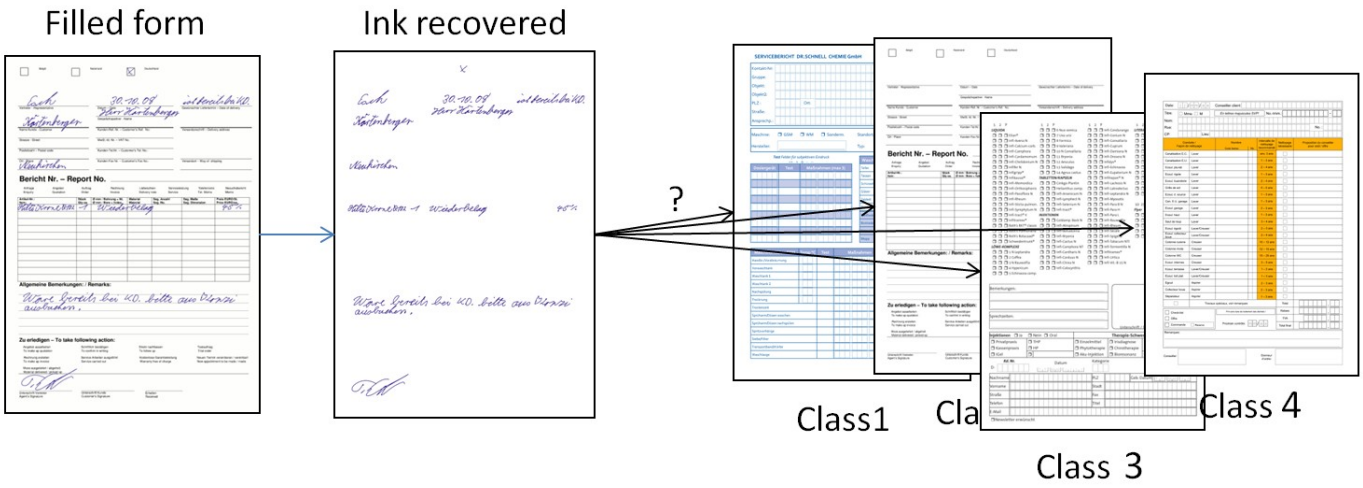


Fig. 4. An overall concept showing note-taking for form classification. In this illustration, the fundamental idea is to select the corresponding form format via ink-tracing files, where four different classes are shown.

by x_v in Eq. (1)) of the corresponding area of interest. To represent it, we simply use value '1' for filled node and '0' , otherwise i.e., empty. The arcs represent the dependencies between the fields. We refer to Section 3 about the way we build BsN. Consider we have a finite set of M forms, in each class $c_{\mathbb{k}}$, $\{f_m\}, m = [1, \ldots, M]$. All classes of the training dataset are represented by a global BN called GBN i.e.,

$$\text{GBN} = \{\text{BN}_{\mathbb{k}}\}_{\mathbb{k}=1\ldots,\mathbb{K}}, \text{ and } \text{BN}_{\mathbb{k}} = \{\text{BsN}_{\mathbb{k}a}\}_{a=1,\ldots,3}, \tag{10}$$

where $c_{\mathbb{k}}$ is composed of three sub-forms (i.e., areas of interest) representing the corresponding BsN.

In what follows, we explain about how training has been made. The training takes place in two stages. Firstly, the main areas corresponding to the most important sub-structures of the form are identified manually and fed into the system to initialise BsN. Secondly, the training is continued for the entire form that connects those variables in BsN. This means that for any field $fl_{\mathbb{k}aq}$, we perform the marginal probability of $p(fl_{\mathbb{k}aq})$ from any sub-form $\text{BsN}_{\mathbb{k}a}$ which is composed of $fl_{\mathbb{k}aq}, q = [1, \ldots, Q]$ fields. The nodes of the graph represent the form fields as well as its class. Within each node, there is a probability distribution which shows the interaction between the nodes. The arcs represent the dependencies between the fields. Once the BsN is trained for all areas of the form with particular distribution probabilities, the training is extended to the entire BN (i.e., GBN) by integrating them. GBN summarises the relationships between the areas representing all form classes. For this, we employ three different algorithms for structure learning: Naive, PC and MWST network, as discussed in Section 3.2.

5 Electronic Note-Taking – Implementation

We remind that the purpose of this study is to classify on-line handwritten forms which are partially filled and the section corresponds to the thorough extension of the previous work [Philippot *et al.*, 2010]. A similar work has been presented [Tran *et al.*, 2010; Hirayama *et al.*, 2011b], from

which the current work has been inspired. Very quickly, overall understanding of the problem is graphically illustrated in Fig. 4. In this illustration, the ink-tracing files from professionals for each sample are used to classify form using the list of provided form formats. The form format is said to be matched with ink-tracing files, if it produces highest similarity (i.e., probability) using BN. The matched format corresponds to the particular class i.e., the form samples are classified.

Our learning is based on what we have extracted as fields and thus field extraction has a direct impact on overall system performance. Therefore, we start with explaining field extraction and then learning. After that we explicitly explain how GBN is learnt from all BsNs. As a reminder, the global concept is of course, based on what we have mentioned in Section 4.

5.1 Field extraction

As said before, field extraction is related to ink-tracing file i.e., a string of coordinates along the pen trajectory from pen-down to pen-up events. This means that we employ matching between the electronic ink-tracing files and the form formats. Our matching is simple and immediate. It is based on whether ink-tracing is found to be in the corresponding box i.e., the proportion of coordinates appeared to be in the corresponding box (cb) with respect to the total number of coordinates in the ink-tracing file (if) i.e.,

$$matching \text{ (if, cb)} = \begin{cases} 1 & \text{if } inside \text{ (if,cb) or } overlapping \text{ (if,cb), and} \\ 0 & \text{if } outside \text{ (if,cb)} \end{cases} \tag{11}$$

We accept the field if the proportion is greater than empirically designed value 0.85% and reject, otherwise. Matching, as a consequence, yields a list of potential filled fields that are then presented in the form of a matrix. In what follows, we will discuss in more detail.

5.2 Learning matrix

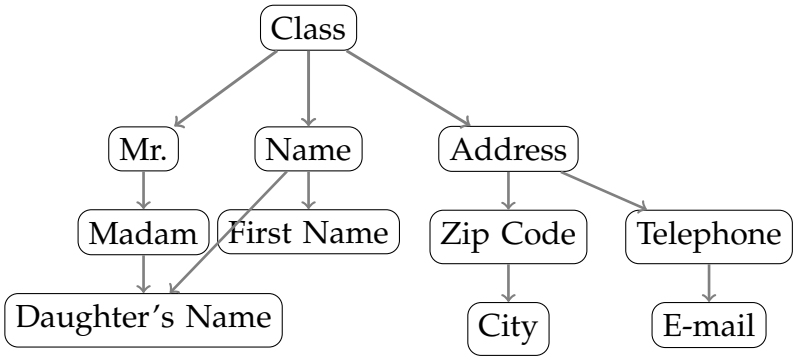
As said before, matrix as a result of matching between ink-tracing files and boxes in form format, are used for learning the BsN. Such a matrix is referred to as learning matrix.

A single matching does not provide sufficient information for learning, however. Therefore, it uses all forms including their format. As a consequence, a learning matrix is obtained for each form format of the particular class. Fig. 5 shows an example of how the matrix is computed. In this table, for each class, three areas of interest are provided including their associated fields. The value '1' in a cell indicates that a single ink-trace (or more) is (are) appeared in the field and hence the field is accepted. The blank cell refers to empty field where not a single trace is matched. From these matrices, we aer able to learn form structure and BsN parameters (*cf.* Section 3.2).

Following Fig. 5, learning matrices (corresponding to three different areas of interest) belonging to any particular class are collectively used for BsN learning. These BsNs are then integrated to build a complete GBN. While discussing BsN and GBN, we put focusing on implmentation issue which is followed by a discussion including illustrations.

Fields	class 1			class 2			class 3			class 4		
Datum TT										1	1	1
Datum MM										1		1
Datum JJ										1		1
Kunderbetereuer								1		1		
Frau												
Herr										1		1
Leig-Nr				1		1				1		
Key										1		
Name	1	1	1	1		1		1		1		1
Srasse	1	1	1	1		1		1		1		1
Hausnr				1	1	1						
PLZ	1	1		1		1			1	1		1
Ort				1				1		1		1

Fig. 5. An example of a learning matrix where three different areas per form class are considered.



		Mr.	
		O	N
Madam	O	0.02	0.88
	N	0.98	0.12

Fig. 6. An example showing a BsN for an *address*.

5.3 BsN Learning

While implementing, we consider the following.

1) We have $\{c_{\mathbb{k}}\}_{\mathbb{k}=1}^{\mathbb{K}}$ classes.
2) Each class $c_{\mathbb{k}}$ is provided with e samples of filled forms and therefore the learning matrix is now composed of $\mathbb{K} \times e$ columns.
3) In each area of interest, we have BsN_{ma} containing Q fields. Now there are $q_{m,a} + 1$ nodes per network, where $q_{m,a}$ is the number of fields in the area a of each class.

To illustrate it, Fig. 6 is an example of the "identification" area i.e., address. The fields are as follows: 1) "Mr.", 2) "Madam", 3) "Daughter's Name", 4) "Name", 5) "First Name", 6) "Address", 7) "Zip Code", 8) "City", 9) "Telephone" and 10) "E-Mail". The BsN is composed of 11 nodes (i.e., 10 + the field class). Concerning the arcs, there is one between "Mr." and "Madam" relating the existence of conditional dependence between them. By observing the probabilities associated with the node "Madam", one can see that the presence of the field "Mr." almost always excludes the presence of the field "Madam". Both the fields can be seen at the same time in the situation where user fills "Mr.", then erases it, and finally checks the field "Madam". In our ink-tracing files, both are considered to be filled. Similarly, both can be absent.

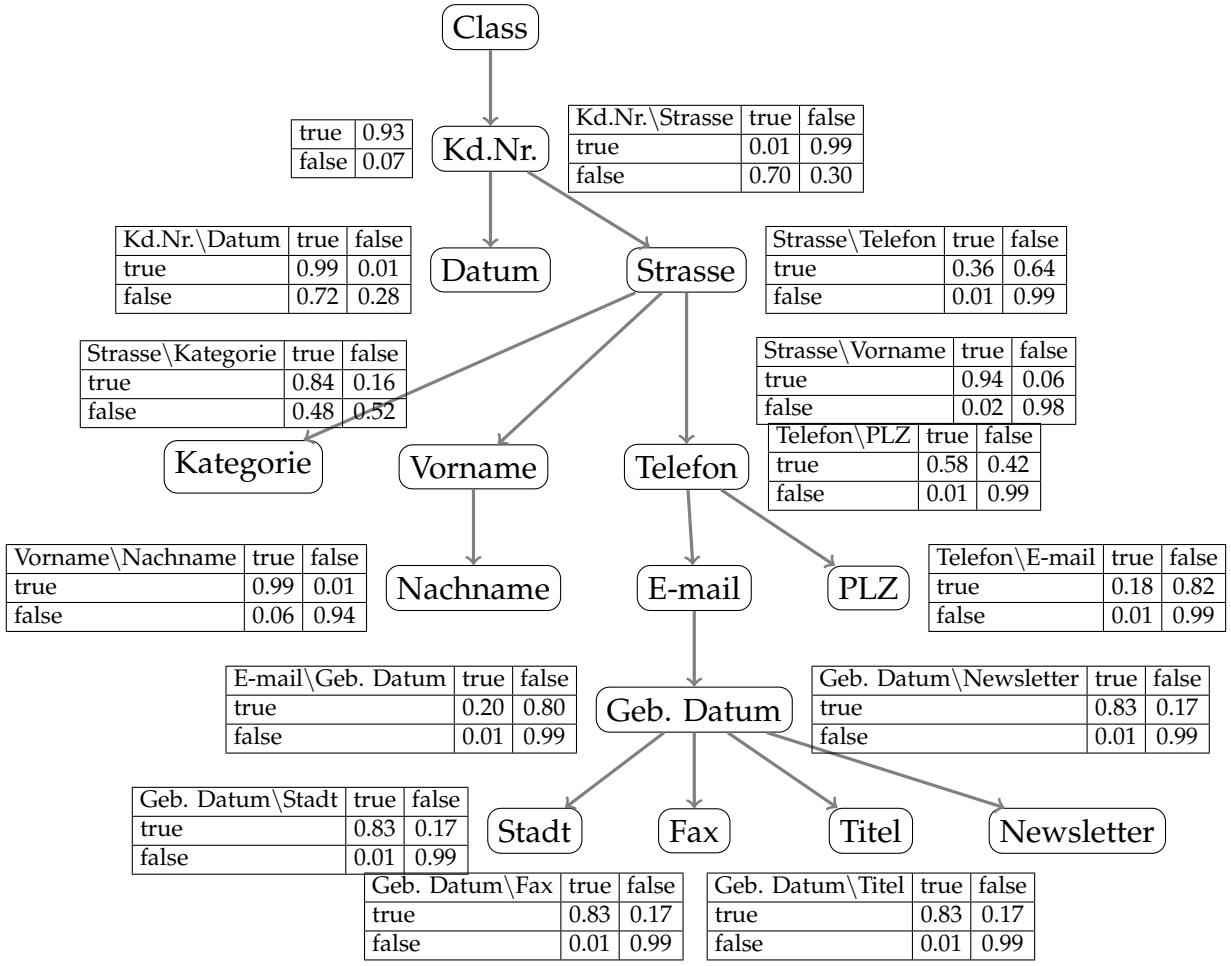


Fig. 7. An example of BsN for form header belonging to class 3, depicted in Fig. 4. It uses PC algorithm.

For further clarity, Fig. 7 shows an example of BsN structure for class 3 (from Table 5) using PC algorithm. There is a link between the filling of "GebDatum" (date of birth) and the "fax number". We will have, for example, a probability of 0.99 having no filled field. In contrast, there is a dependence between "Vorname" and "Nachname" (respectively first and last name) that seems more logical and intuitive. We will find a probability of 0.99 that both fields are filled simultaneously. Besides, we have received similar behaviour while using the remaining two algorithms: Naive and MWST. However, their structures are different from each other. In all three algorithms, we compute the maximum likelihood estimation to estimate parameter.

5.4 GBN learning

We repeat that three different BsNs are now used to build a GBN. This means that it is required to create new learning matrix from BsNs. To efficiently handle it, we consider mainly two steps: data creation and its discretisation.

Data creation.
From each BsN, we compute the marginal probabilities of each area using local networks of

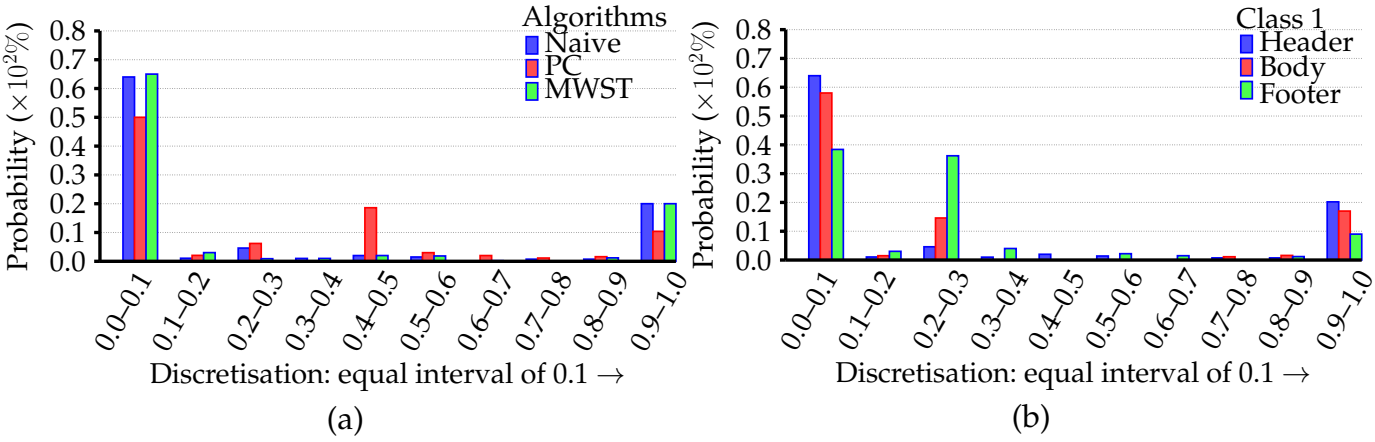


Fig. 8. Probability distribution using form class 1: (a) for form headers when all three learning algorithms are used and (b) for three different areas of interest when Naive BN is used.

the defined area (i.e., header, body and footer) via inference. Since we have several classes and forms (due to similar structure, for instance), we receive sveral different probabilities.

1) For m forms of three different areas distributed in $\mathbb{K}$ classes, there are $m \times 3 \times \mathbb{K}$ probabilities distributed in a learning matrix having $3 \times \mathbb{K} + 1$ columns in m lines.
2) Now, the number of columns this new matrix will correspond to the number of computed probabilities for a single form plus its class.

Discretisation.

The obtained matrix is composed of probabilities in the interval $[0, 1]$. To make possible BsNs integration into a GBN, we discretise them in the interval $[0, 1]$ in two different ways.

1) Equal intervals.
   An equal interval of 0.1, for instance can be used i.e., $[0, 0.1]$, $]0.2, 0.3]$, $\ldots$, $]0.9, 1]$.
2) Unequal intervals.
   One can, for example use the following five intervals: $[0, 0.05]$, $]0.05, 0.1]$, $]0.1, 0.9[$, $[0.9, 0.95[$ and $[0.95, 1]$.

In each case of discretisation, several tests have been made. In the following, to provide an intuitive feelings about data creation and discretisation, we attempt to guide readers with a couple of illustrations, from a single class (class 1).

1) Probability distribution using all learning algorithms.
   To be more specific, we take form header belonging a single class. It is shown in Fig. 8(a). In this illustration, it is found that the distribution is fairly uniform from all learning algorithms. There exists, however a peak of 20% for the PC algorithm for probabilities belonging to the interval $]0.4, 0.5]$. This means that 20% of forms, the algorithm PC generates ambiguity during the classification. While, more than 65% of the forms are lying in the probability range of 0 and 0.1, according to the MWST and Naive. Knowing that 75% of the forms do not belong to class 1, which indicates that for less than 10%, there remains a confusion since we assume that there exists no classification error.
2) Probability distribution in three different areas of the form using the Naive BN.

For more deeper analysis, the graph in Fig. 8(b) gives the probability distribution for the three areas of the form class using the Naive BN. As before, we observe two peaks on the intervals $[0, 0.1[$ and $[0.9, 1]$. However, the classification of the footer does not provide significant separation. It happens because there exists always a freedom of writing.

Based on the observation from two different cases presented in Fig. 8, one can notice that similar behaviour will be shown for all other remaining classes. This will be evaluated using precision and recall, while doing recognition (*cf.* Section 5.5). Similarly, unequal discretisation can affect the performance of the proposed approach. In Section 6, we will provide recognition performance of the proposed approach. While reporting recognition performance, based on our experience, we will use equal interval discretisation because of the better efficacy.

5.5 Form recognition

As in learning phase, recognition is done in two steps:

1) the recognition of areas and
2) global recognition.

In order to speed up the process, we first check whether separate BsN (from each area) is matched. This goes to all areas one by one. If they are matched, we go for complete GBN matching. Bullet-wise summary can be explained as follows.

1) We first, use BsNs for areas using probability inference i.e., for a form f_m having three areas of interest in $\mathbb{K}$ classes, we calculate the probabilities.
2) For each area of interest, we then have a column matrix of $\mathbb{K} \times \mathbb{K}$ probabilities.
3) For the recognition of the entire form, we discretise this matrix as in learning.
4) From the discretised matrix and the GBN, we then seek the most similar class via inference. The most similar class is said to be recognised if it matches the ground-truths.

6 Experiments

6.1 Dataset and evaluation protocol

To validate our approach, we conducted a series of tests over four different classes, each consisting of 800 forms. These samples were filled for commercial and administrative purposes.

For recognition performance evaluation, we are based on the following standard metrics:

1) precision and recall, and
2) F-score.

Formally, precision is the fraction of retrieved documents that are relevant to the search i.e.,

$$\text{precision} = \frac{\text{relevant documents} \bigcap \text{retrieved documents}}{\text{retrieved documents}}. \tag{12}$$

Precision takes all retrieved documents into account, but it can also be evaluated at a given cut-off rank, considering only the top-most results returned by our system. This means that precision is the ratio of number of correct documents divided by the number of all returned documents for a given short-list. Note that precision is also used with recall, the percent of all

relevant documents that is returned by the search. Recall is the fraction of the documents that are relevant to the query that are successfully retrieved i.e.,

$$\text{recall} = \frac{\text{relevant documents} \bigcap \text{retrieved documents}}{\text{relevant documents}}. \tag{13}$$

The traditional F-measure or balanced F-score (i.e., F_1-score) is the harmonic mean of precision and recall,

$$F_1\text{-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{14}$$

The F_1-score is the interpreted result of the weighted average of the precision and recall. F_1-score reaches its best value at 1 and worst score at 0.

Besides, computing ROC curve is still interesting. For any dataset, since the ROC curve and the precision-recall (PR) curve for a given algorithm contain the same points, we do not need to include ROC curve in our test. In other words, the idea leads to the theorem that a curve dominates in ROC space if and only if it dominates in PR space [Davis and Goadrich, 2006].

Unlike traditional dichotomous classification of database, $\mathcal{K}$-fold cross validation (CV) has been implemented. In $\mathcal{K}$-fold CV, the original database for every class is randomly partitioned into $\mathcal{K}$ sub-database. Of the $\mathcal{K}$ sub-database, a single sub-database is used for validation, and the remaining $\mathcal{K} - 1$ sub-databases are used for learning. This process is then repeated for $\mathcal{K}$ folds, with each of the $\mathcal{K}$ sub-databases used exactly once. Eventually, a single value results from averaging all. In our case, $\mathcal{K} = 4$, where we have 600 samples for learning and the remaining 200 samples for testing from each class.

6.2 Results and analysis

We perform a series of tests to validate the proposed approach. As said before (*cf.* Section 5.4 in pp. 15), the results will be reported by using equal interval of discretisation.

In the very beginning, the test has been made for different learning sizes of the datasets. This means that, to validate learning, it makes sense to evaluate the outputs where we aim to see whether size of the learning dataset affects the performance of the approach. As said before, we have 600 learning samples from each class. Rather than using all samples for learning, our learning has been started from 50 samples per class. Fig. 9 shows results precision and recall from three different learning algorithms over different dataset sizes. In Fig. 9, MWST, provides better performance in comparison to others. Naive and PC show almost similar behaviour. Without a surprise, the larger the dataset, the higher the recognition performance. This is one of the common characterisation of all three algorithms. This means that as soon as learning dataset size increases, structure of the BsN is updated with additional relevant fields associated with it. Fig. 10 shows an example of it.

For the remaining tests, we provide algorithm-wise results. To simplify it, we first compute separate recognition performance using precision and recall, from each area of interest on a one-to-one basis. These performances are then averaged. Finally, we analyse the performances from one learning algorithm to another.

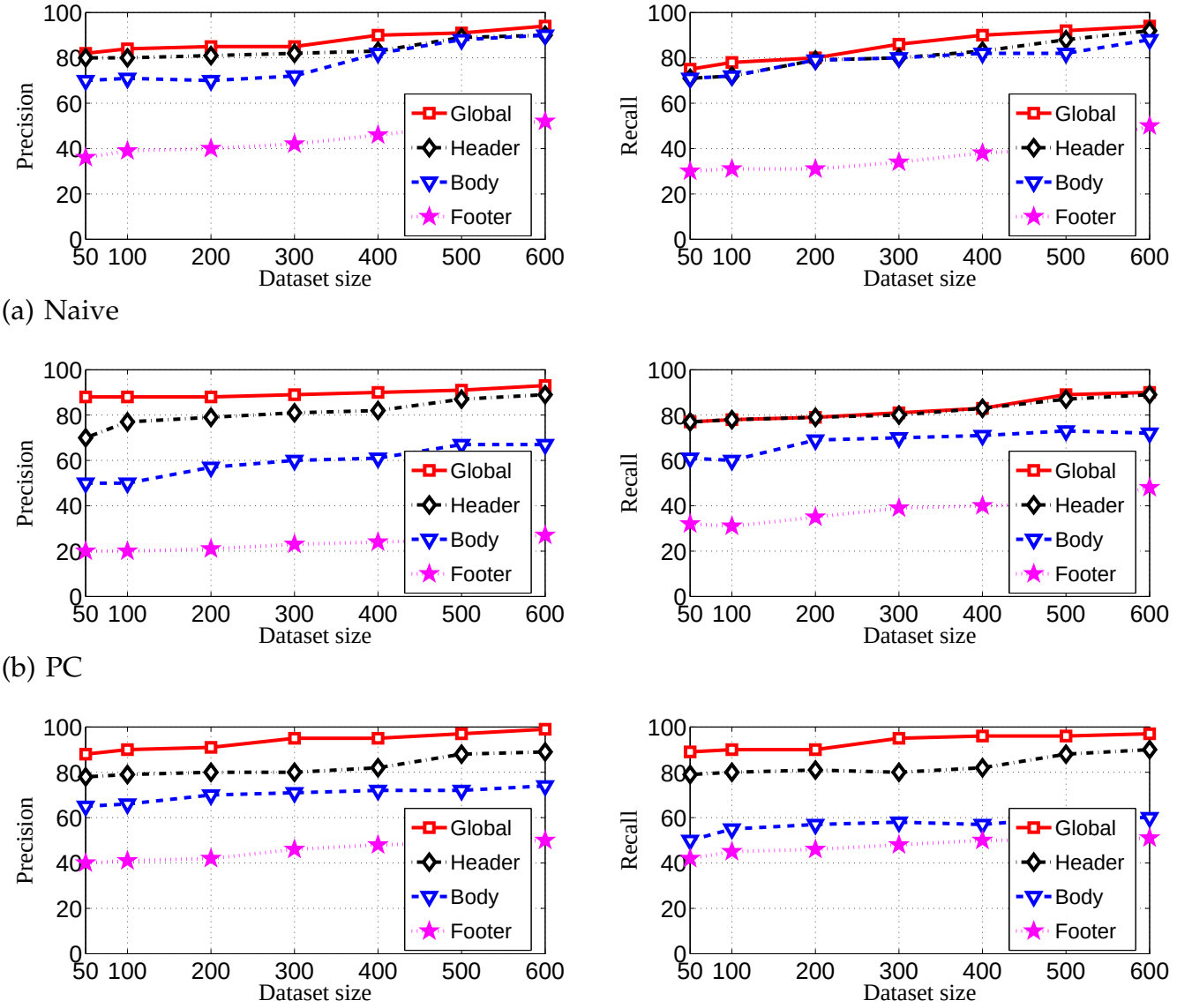
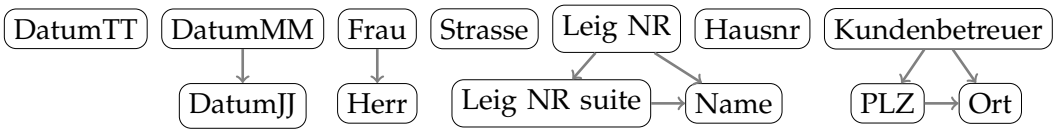


Fig. 9. Precision and recall from three different learning algorithms over different dataset sizes.

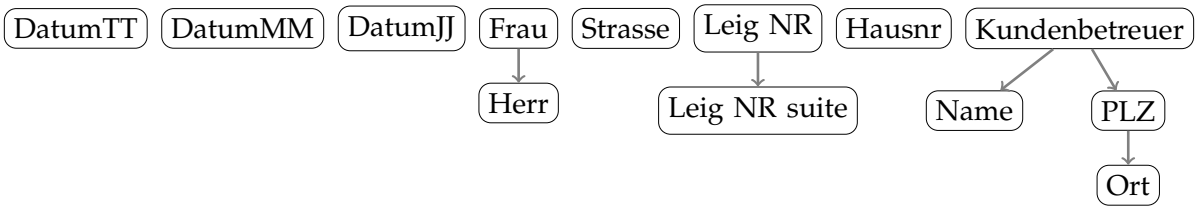
Naive.

To provide clarity about the tests, we provide area-wise performance using Naive BN.

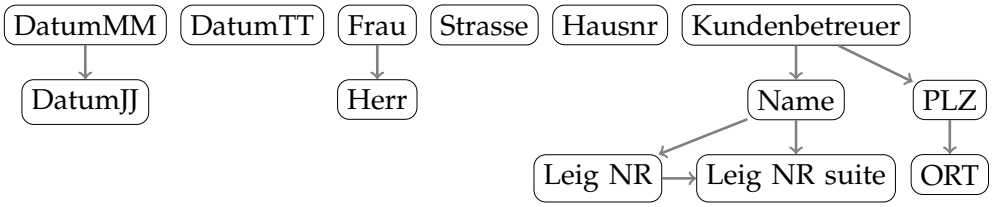
1) Form headers
   In Table 1, precision and recall using Naive BN is provided for form headers. It uses four different learning datasets. The results show a correlation between the training set and the results, and thus the network parameters differ from one dataset to another.
2) Form bodies
   Table 2 gives the results for form bodies. Compared to the performance for headers (*cf.* Table 1), it provides less recognition scores. This is primarily due to the large number of fields, overlapping (sometimes) with the fields belonging to headers.
3) Form footers
   Table 3 gives the results for form footers. It shows quite less test scores in comparison to



(a) BsN structure using 50 samples.



(b) BsN structure using 100 samples.



(c) BsN structure using 400 samples.

Fig. 10. An example of showing BsN structure update, for the header of the class 4, in particular, starting from 50 to 400 samples with the PC algorithm.

TABLE 1
Precision, recall and F_1-score (in %) for the form headers using Naive BN.

	Precision				Recall				F_1-score			
Dataset$_\Rightarrow$	1	2	3	4	1	2	3	4	1	2	3	4
class 1	86.40	83.19	87.95	88.34	98.50	99.00	98.50	98.50	92.05	90.41	92.92	93.14
class 2	82.30	93.90	83.33	82.84	98.80	98.70	99.60	99.66	89.79	96.24	90.74	90.47
class 3	98.80	98.64	98.60	99.80	61.50	76.14	64.40	64.50	75.81	85.94	77.91	78.35
class 4	96.35	99.50	97.07	95.15	99.20	98.16	99.50	98.34	97.75	98.82	98.27	96.71
Average	90.96	93.81	91.74	91.53	89.50	93.00	90.50	90.25	90.22	93.40	91.11	90.88

headers and bodies, because the footers are composed of a few fields. For example, class 4 comprises of only three fields that are free enough at making-notes. As a consequence, classification is not guaranteed.

4) Global form
Table 4 shows the results for the whole forms. It provides satisfactory results, in overall. Unlike the previous results (*cf.* Table 3) for footers, global form classification receives better results. This means that global form recognition does not show trade-off behaviour of the recognition performance even when form footers are not clearly separated during classification.

PC and ***MWST***.
Previously, we have provided area-wise recognition performance. This is aimed to provide

TABLE 2
Precision, recall and F_1-score (in %) for the form bodies using Naive BN.

	Precision				Recall				F_1-score			
Dataset$_\Rightarrow$	1	2	3	4	1	2	3	4	1	2	3	4
class 1	97.03	99.00	98.01	98.01	98.00	99.00	98.50	98.50	97.51	99.00	98.25	98.25
class 2	67.82	74.33	63.23	66.11	98.00	97.00	98.00	99.50	80.16	84.17	76.87	79.44
class 3	96.04	93.89	94.81	98.81	48.50	62.80	36.50	41.84	64.45	75.26	52.71	58.79
class 4	95.19	96.63	93.87	93.46	99.00	99.20	99.50	99.66	97.06	97.90	96.60	96.46
Average	89.02	90.96	87.48	89.10	85.88	89.50	83.13	84.88	87.42	90.22	85.25	86.94

TABLE 3
Precision, recall and F_1-score (in %) for the form footers using Naive BN.

	Precision				Recall				F_1-score			
Dataset$_\Rightarrow$	1	2	3	4	1	2	3	4	1	2	3	4
class 1	96.43	89.64	95.65	91.52	78.00	79.00	77.00	81.00	86.24	83.98	85.32	85.94
class 2	34.21	39.29	31.85	32.68	98.43	99.75	99.67	99.87	50.77	56.37	48.27	49.25
class 3	23.34	23.48	98.77	98.80	02.55	13.50	00.50	00.50	04.60	17.14	00.99	00.99
class 4	04.54	00.13	01.23	02.25	04.50	00.25	00.33	00.13	04.52	00.17	00.52	00.25
Average	57.13	38.14	56.87	56.31	45.87	48.13	44.38	45.38	50.88	42.56	49.85	50.26

deeper analysis of the results. For PC and MWST, unlike before, we provide precision and recall for global forms. Table 5 and 6 provide results respectively from PC and MWST. Any of the two performs better than Naive BN, presented before. In overall comparison, MWST performs better than PC, showing marginal difference of not more than 1% recognition rate.

On the whole, our results provide the fact that local areas do not really affect the global form recognition. Such a behaviour has been received from all learning algorithms.

7 Discussions

We have addressed the problem of recognising handwritten forms from incomplete and uncertain electronic ink-tracing files. It is therefore necessary to rely on the presence or absence of ink files in the fields so that we are able to relocate the position whether the physical structure of the form recognition is possible. In this framework, we take advantage of BNs to handle such incomplete

TABLE 4
Precision, recall and F_1-score (in %) for the global forms using Naive BN.

	Precision				Recall				F_1-score			
Dataset$_\Rightarrow$	1	2	3	4	1	2	3	4	1	2	3	4
class 1	90.91	91.19	92.95	98.18	99.02	99.25	99.33	99.00	94.79	95.05	96.03	98.59
class 2	90.98	92.02	90.63	84.88	99.75	99.96	99.60	99.67	95.16	95.83	94.90	91.68
class 3	99.25	98.99	99.50	99.75	80.23	82.79	82.74	81.50	88.73	90.17	90.35	89.71
class 4	99.50	99.49	99.67	99.50	99.67	98.33	99.50	99.33	99.58	98.91	99.58	99.41
Average	95.16	95.42	95.69	95.58	94.67	95.08	95.29	94.88	94.91	95.25	95.49	95.23

TABLE 5
Precision, recall and F_1-score (in %) for the global forms using PC BN.

	Precision				Recall				F_1-score			
Dataset$_\Rightarrow$	1	2	3	4	1	2	3	4	1	2	3	4
class 1	95.24	57.14	68.72	99.70	99.50	99.75	97.50	90.75	97.32	72.66	80.62	95.01
class 2	99.60	99.90	99.50	99.40	99.67	87.83	80.00	99.70	99.63	93.48	88.69	99.55
class 3	99.75	99.67	90.91	93.60	97.09	99.67	99.25	99.00	98.40	99.67	94.90	96.22
class 4	97.50	99.99	99.67	97.90	97.50	37.50	67.60	99.90	97.50	54.54	80.56	98.89
Average	98.02	89.18	89.70	97.65	98.44	81.19	86.09	97.34	98.23	85.00	87.86	97.49

TABLE 6
Precision, recall and F_1-score (in %) for global forms using MWST BN.

	Precision				Recall				F_1-score			
Dataset$_\Rightarrow$	1	2	3	4	1	2	3	4	1	2	3	4
class 1	98.83	98.50	98.67	98.67	98.02	97.95	96.42	96.42	98.42	98.22	97.53	97.53
class 2	99.88	99.77	97.60	98.50	99.67	99.80	99.01	98.36	99.77	99.78	98.30	98.43
class 3	97.45	97.90	87.77	95.00	97.99	97.44	99.61	97.91	97.72	97.67	93.32	96.43
class 4	99.17	98.33	99.80	99.33	99.66	99.33	89.96	98.84	99.41	98.83	94.62	99.08
Average	98.83	98.63	95.96	97.88	98.83	98.63	96.25	97.88	98.83	98.63	96.10	97.88

data where the use of conditional probabilities can highlight the relationships between random variables and inference can deal with unobserved data. In our case, the random variables refer to fields and parts of forms and the unobserved data refer to form class in addition to fields and parts. For validation, we have studied three fundamental types of BNs: Naive, PC and MWST where each learning structure uses different assumptions of independence. Besides, we also reported the better discretisation, by considering the efficacy of the system.

We have observed that the conditional probabilities of BNs provide the contextual relationships and circumstances. To limit the context and enhancing their impact, three different areas of interests i.e., header, body and footer have been employed. Based on the reported results, we found that whatever the algorithm used, the body is the best recognised area since it contains sufficient fields and thus more contextual relationships between them. Basically, this happens due to the presence of tables which are interconnected with a very high chance of filling fields. Among them, MWST provides an accuracy up to 92.47%. In contrast, the worst results have found in the footer area. This is primarily because of the presence of some fields that tend to overlap from one form to another.

Overall, our system shows satisfactory performance in the real-world industrial problem, where separate (areas of interest wise) information are exploited via BsNs based on uncertain and incomplete data.

8 Conclusion and Future Perspectives

In this paper, we have used Bayesian networks for form recognition where fields are partially filled using electronic ink. BN exploits the possible relationships between corresponding fields

in conditional probabilities. For simplicity, forms are first split into specific areas and built Bayesian sub-network (BsN) per area. These BsNs are then integrated to construct a global BN, representing whole form. For learning, we have studied three major algorithms: Naive, PC and MWST. For validation, we have taken a real-world electronic note-taking application. Among the three learning algorithms and considering our application, we have found that the MWST provides better performance.

Based on the classical learning algorithms, we have established the interests of the technique. In this framework, use of advanced and optimized learning algorithms like super parent TAN (SP-TAN) algorithm [Keogh and Pazzani, 1999], Averaged one-dependence estimators (AODE) [Webb *et al.*, 2005], hidden Naive Bayes (HNB) [Jiang *et al.*, 2009], and discriminatively weighted Naive Bayes (DWNB) [Jiang *et al.*, 2012] would improve the results. Combining these different structures of BNs would be another interesting plan to go further since not a single algorithm can overcome for all local areas when recognition performance is taken into account. Within this framework, one could imagine a system that automatically choose the network that gives the best results based on data to be processed.

Conflict of Interest

None Declared.

Author contributions

Ms. E. Philoppot performed experiments, under the supervision of Dr. A. Belaïd and Dr. Y. Belaïd. Following the draft made by Ms. E. Philoppot and Dr. A. Belaïd, Dr. K.C. Santosh analysed data and results, and wrote a complete paper and responses to the anonymous reviewers in addition to the supplementary results.

References

Belaïd, A. (2001). Recognition of table of contents for electronic library consulting. *International Journal on Document Analysis and Recognition*, **4**(1), 35–45.

Cho, S.-J. and Kim, J. H. (2003). Bayesian network modeling of hangul characters for on-line handwriting recognition. In *Proceedings of the IAPR International Conference on Document Analysis and Recognition*, pages 207– 2011.

Chow, C. and Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, **14**(3), 462–467.

Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the International Conference on Machine Learning*, pages 233–240. ACM.

Denoyer, L. and Gallinari, P. (2004). Bayesian network model for semi-structured document classification. *Inf. Process. Manage.*, **40**(5), 807–827.

François, O. and Leray, P. (2006). Learning the tree augmented naive bayes classifier from incomplete datasets. In *Proceedings of European Workshop on Probabilistic Graphical Models*, pages 91–98.

Friedman, N. and Goldszmidt, M. (1996). Building classifiers using bayesian networks. In *Proceedings of the national conference on Artificial intelligence - Volume 2*, AAAI'96, pages 1277–1284.

Friedman, N., Geiger, D., Goldszmidt, M., Provan, G., Langley, P., and Smyth, P. (1997). Bayesian network classifiers. **29**, 131–163.

Hallouli, K., Likforman-Sulem, L., Sigelle, M., and Sigelle, M. (2002). A comparative study between decision fusion and data fusion in markovian printed character recognition. In *Proceedings of the IAPR International Conference on Pattern Recognition*, pages 147–150.

He, Y.-L., Wang, R., Kwong, S., and Wang, X.-Z. (2014). Bayesian classifiers based on probability density estimation and their applications to simultaneous fault diagnosis. *Information Sciences*, **259**, 252–268.

Hirayama, J., Shinjo, H., Takahashi, T., and Nagasaki, T. (2011a). Development of template-free form recognition system. In *Proceedings of the IAPR International Conference on Document Analysis and Recognition*, pages 237–241.

Hirayama, J., Shinjo, H., Takahashi, T., and Nagasaki, T. (2011b). Development of template-free form recognition system. In *Proceedings of the IAPR International Conference on Document Analysis and Recognition*, pages 237–241.

Jensen, F. V. (1996). *Introduction to Bayesian Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition.

Jensen, F. V., Lauritzen, S. L., and Olesen, K. G. (1990). Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quaterly*, **4**, 269–282.

Jiang, L., Zhang, H., Cai, Z., and Su, J. (2005). Learning tree augmented naive bayes for ranking. In L. Zhou, B. Ooi, and X. Meng, editors, *Database Systems for Advanced Applications*, volume 3453 of *Lecture Notes in Computer Science*, pages 688–698. Springer Berlin Heidelberg.

Jiang, L., Wang, D., and Cai, Z. (2007). Scaling up the accuracy of bayesian network classifiers by m-estimate. In D.-S. Huang, L. Heutte, and M. Loog, editors, *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, volume 4682 of *Lecture Notes in Computer Science*, pages 475–484. Springer Berlin Heidelberg.

Jiang, L., Zhang, H., and Cai, Z. (2009). A novel bayes model: Hidden naive bayes. *IEEE Transactions on Knowledge and Data Engineering*, **21**(10), 1361–1371.

Jiang, L., Wang, D., and Cai, Z. (2012). Discriminatively weighted naive bayes and its application in text classification. *International Journal on Artificial Intelligence Tools*, **21**(1).

Jiang, L., Cai, Z., Wang, D., and Zhang, H. (2013). Bayesian citation-knn with distance weighting. *International Journal of Machine Learning and Cybernetics*, pages 1–7.

Kebairi, S., Taconet, B., Zahour, A., and Ramdane, S. (1998). A statistical method for an automatic detection of form types. In *Proceedings of International Workshop on Document Analysis Systems*, pages 84–98.

Keogh, E. and Pazzani, M. (1999). Learning augmented bayesian classifiers: A comparison of distribution-based and classification-based approaches. In *Proceedings of the seventh international workshop on artificial intelligence and statistics*, pages 225–230.

Langley, P., Iba, W., and Thompson, K. (1992). An analysis of bayesian classifiers. In *AAAI*, pages 223–228.

Likforman-Sulem, L. and Sigelle, M. (2008). Recognition of degraded characters using dynamic bayesian networks. *Pattern Recognition*, **41**(10), 3092–3103.

Likforman-Sulem, L. and Sigelle, M. (2009). Combination of dynamic bayesian network classifiers for the recognition of degraded characters. In *Proceedings of the SPIE International Symposium on Document Recognition and Retrieval*, pages 1–10.

Mahjoub, M. A. and Jayech, K. (2010). ndexation de structures de documents par rseaux baysiens. pages 163–178.

Naïm, P. W., P. Leray, O. P., and Becker, A. (2007). *Réseaux bayésiens*. Eyrolles.

Neapolitan, R. (2004). *Learning Bayesian Networks*. Prentice Hall, Upper Saddle River, NJ.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Philippot, E., Belaïd, Y., and Belaïd, A. (2010). Bayesian networks learning algorithms for online form classification. In *Proceedings of the IAPR International Conference on Pattern Recognition*, pages 1981–1984.

Piwowarski, B., Denoyer, L., and Gallinari, P. (20002). Un modle pour la recherche d'information sur des documents structurs. In *Journes internationales d'Analyse statistique des Donnes Textuelles (JADT)*.

Russell, S. J. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education.

Santosh, K. C., Nattee, C., and Lamiroy, B. (2012). Relative positioning of stroke-based clustering: a new approach to online handwritten devanagari character recognition. *International Journal of Image & Graphics*, **12**(2), 1250016.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, pages 1–47.

Souafi-Bensafi, S., Parizeau, M., Lebourgeois, F., and Emptoz, H. (2002). Bayesian networks classifiers applied to documents. In *Proceedings of the IAPR International Conference on Pattern Recognition*, page 483.

Spirtes, P., Glymour, C., and Scheines, R. (2001). *Causation, Prediction, and Search*. The MIT Press, Cambridge, MA, USA, second edition.

Subrahmanya, N. and Shin, Y. (2013). A variational bayesian framework for group feature selection. *International Journal of Machine Learning and Cybernetics*, **4**(6), 609–619.

Tran, D. C., Franco, P., and Ogier, J.-M. (2010). Form recognition from ink strokes on tablet. In *Proceedings of International Workshop on Document Analysis Systems*, pages 293–300.

Verron, S., Tiplica, T., and Kobi, A. (2007). Multivariate control charts with a bayesian network. In *ICINCO-ICSO*, pages 228–233.

Wang, X.-Z., He, Y.-L., and Wang, D. (2014). Non-naive bayesian classifiers for classification problems with continuous attributes. *IEEE Transactions on Cybernetics*, **44**(1), 21–39.

Webb, G. I., Boughton, J. R., and Wang, Z. (2005). Not so naive bayes: Aggregating one-dependence estimators. *Machine Learning*, **58**(1), 5–24.

Weissenbacher, D. (2006). Bayesian network, a model for nlp? In *Eleventh Conference of the European Chapter of the Association for Computational Linguistics*, EACL, pages 195–198.

Weissenbacher, D. and Nazarenko, A. (2011). Understand the effects of erroneous annotations produced by nlp pipelines, a case study on the pronominal anaphora resolution. *Traitement Automatique des Langues*, **52**(1), 161–185.

Wong, M. L. and Leung, K. S. (2004). An efficient data mining method for learning bayesian networks using an evolutionary algorithm-based hybrid approach. *IEEE Transactions on Evolutionay Computation*, **8**(4), 378–404.